Remarks

The Office Action dated November 6, 2003, which was a final rejection has been noted and its contents carefully studied. Responsive to this Office Action, there is concurrently being filed herewith a Request for Continued Examination, which is further supported by the amendments set forth herein. In light of the amendments to the claims and arguments submitted, reconsideration and withdrawal of the rejection under 35 U.S.C. §102 and §103 is courteously requested.

It is noted that the three month date for Response or Appeal in the Final Rejection was February 6, 2004. This Amendment and RCE are being filed no later than April 6, 2004, along with a Petition of Two Month Extension of Time and corresponding payment, setting April 6, 2004 as the due date for the filing of this Amendment and Petition for Extension of Time. Accordingly, the filing of this Amendment is timely and proper.

Turning to the amendments made herein, it is noted that claims 1-3 and 5-13 have been cancelled. New claims 16-18 have been added which are in independent claim form. This leaves just six claims, all in independent claim form.

New claim 16 corresponds in independent form to the features of claims 1-3 combined. New claim 17 corresponds to the features of claims 5-9 combined in independent claim form. New claim 18 corresponds to the features of claims 10-13 combined in independent claim form. Claims 4, 14 and 15 were previously amended and are again presented herein for reconsideration.

With respect to new claim 16, the invention claims a method of searching a file access system for a requested file. The steps of the method include establishing a field in a directory i-node memory structure for files corresponding to a directory cache hash table. The field contains a pointer to the directory cache hash table. Memory is allocated for a directory cache and buffer cache hash table having an array of hash buckets which point to a list of files which may correspond to a specific i-node. The directory cache hash table stores directory layouts and the allocating of memory for the directory cache hash table is done in a manner to include selecting

directories to cache using at least one of the number of files in a directory and the frequency of use of the files. The directory cache hash table is searched for a requested file by hashing the file i-node to a specific bucket which contains a list of files that may correspond to the requested file i-node. If the file name in the directory cache hash table is not found, then the file structures are conventionally searched. If, on the other hand, the bucket contains a matching file name, then the system points towards the name of the requested file as stored.

Claim 17 is similar to claim 16, but further includes the step of converting the directory to a faster representation. The faster representation includes a pointer from the directory i-node to an associated hash table. As before, the hash table contains a layout of the directory with an array of hash buckets which point to a list of files which may correspond to a specific i-node. In this case, hashing of selected directories into a hash table format is done according to at least one of: (1) size of the directory, (2) frequency of access, and (3) a user selected criteria.

Claim 18 is similar to claim 16 except that instead of allocating memory, a hash table is allocated which has hash buckets which point to a list of files which may correspond to a specific i-node. Yet further, hash buckets are linked to offsets where a name of a requested file is stored and a pointer is established for the directory, with the pointer pointing from a directory i-node to the hash table, and thereafter searching the hash buckets for a requested file.

Claims 4, 14 and 15 have been previously discussed and need not be restated in detail herein. The Examiner is kindly invited to the previously submitted discussion of claims 4, 14 and 15.

It is noted that the cited references have been previously discussed in the original Response to the Final Rejection.

A further review of the references suggests, and Applicants kindly urge, that the Examiner may be misinterpreting the teachings of the references in the context of the claimed invention, and that the invention as set forth in the claims is clearly patentable under 35 U.S.C. §102 and/or §103 over the cited references. Thus, a more thorough detailed discussion of the references is further presented herein for the Examiner's kind and consideration.

U.S. Patent 5,151,989 to Johnson et al.

U.S. Patent No. 5,151,989 to Johnson et al. is the main reference cited by the Examiner along and/or in combination with other references to reject the claims. In rejecting previous claims 1-2, 4-6, 10 and 13-15, the Examiner asserted that Johnson teaches a method of searching a file access system for a requested file (citing column 8, lines 9-22 and the Abstract). The Johnson patent is asserted to teach establishing a field in a directory i-node memory structure for files corresponding to a directory cache hash table, the field containing a pointer to the directory cache hash table (asserted to be taught by column 13, lines 52-59; column 17, line 64 - column 18, line 8; column 20, lines 63 – column 21, line 11; column 18, lines 17-22; and column 22, lines 17-22; and column 22, lines 35-48). The Examiner further cites other portions of the patent as teaching the recited allocating memory for directory cache, searching the directory cache hash table and if the bucket contains a matching file name pointing toward the name of where the requested file is stored as recited in Applicants' claims. It is respectfully urged that this is not what is actually taught by the patent in the specifically cited sections. In this context, the actual teachings of the patent are further restated herein for the Examiner's kind consideration.

What Johnson teaches is a directory caching technique for plurality of data processing systems which are connected together in a network. Johnson provides that when a local, or client, data processing system interrogates a remote, or server, data processing system for a unit of directory information, the server system is enabled to automatically send additional units of pertinent directory information back to the client system in response to a subsequent change in the directory structure of the server system (column 5, lines 52-68). In one aspect, the essential aspects of the file system that are relevant are that each file on an individual file system is uniquely identified by its i-node number, and that directories are files and thus a directory can be uniquely identified by its i-node number.

In the file system independent portion of the system, whenever an i-node needs to be represented, it is represented by a v node. This has nothing to do with the claimed invention. The elements in the structure include a pointer to the vfs which contains the v node, a pointer to the vfs which is mounted over the i node, a pointer to either a surrogate i node or a real i node,

and a pointer to the node table entry. The essential elements of an i node table entry are a pointer to the head of a file access structure list and information from the disk i node. As before, this has nothing to do with the claimed invention as specifically recited in the claims. More particularly, this has nothing to do with establishing a field in a directory i-node memory structure for files corresponding to a directory cache hash table, with the field containing a pointer to the directory cache hash table.

The later section cited by the Examiner with reference to column 13, deals with a completely different aspect of operation of the system of Johnson and is not related to the earlier section discussed above. More specifically, this discussion has to do with opening files by a remote node which has an i node table entry in the server's i node table. More specifically, when a remote lookup request occurs, the file is identified by a file handle. The file handle contains the fields of: device number, i node number, and i node generation number. The i node generation number stored on disk is a field in the i node. When the server deletes a file it increments the i node generation number. Clearly this has nothing to do with the claimed invention. While each node is indicated as containing a node table, the node uses entries in its node table to record information about existing connections to remote nodes, something having nothing to do with the claimed invention.

Johnson teaches defining a hash function that hashes any node ID, directory file handle, and path name part into an index into a hash table. In contrast, Applicants' establishes a field in a directory i-node memory structure, containing a pointer to a directory cache hash table. The directory cache hash table has a array of hash buckets which point to a list of files which may correspond to the specific i-node, and the directory cache hash table stores directory layouts. The step of allocating memory for the directory cache has table includes selecting directories to cache using at least one of the number of files in a directory and the frequency of use (claim 16), the cache hash table as searched and if the file name is not found, conventionally searching file structures. If the file name is found, the system then points to where the name of the requested file is stored (claim 16). Accordingly, it is respectfully urged that the claimed invention is not anticipated or obvious under 35 U.S.C. §102 and/or 35 §103 in light of the cited reference

standing alone, or in combination with the other references which are discussed hereafter for the Examiner's further consideration.

U.S. Patent 5,666,532 to Saks et al

U.S. Patent 5,666,532 to Saks et al (hereinafter Saks) merely teaches the conventional concept of buffering the block device model to reduce I/O traffic. The size of buffers used in performing transfers from or to block devices are some convenient multiples of 512, which is the size of a typical disk block. The buffering for block I/O data transfers is performed either by using some or all pages from a page cache, or by a dedicated memory of pool called buffer cache, and frequently by employing both the page cache and a buffer cache.

While teaching generally allocating memory for buffers, this is not the same as allocating memory for a directory cache and buffer cache hash table having an array of hash buckets which point to a list of files which may correspond to a specific i-node. Moreover, there is no discussion of how directories to cache are selected including the using of at least one of the number of files in a directory and/or the frequency of use as is recited in Applicants' claimed invention. Thus, Saks adds nothing to the teachings of Johnson. Moreover, it is also urged that one of ordinary skill in the art would not look to the teachings of Saks to modify Johnson. More specifically, Johnson teaches directory cache management in a distributed data processing system whereas Saks teaches a method for a synchronous ordered operation in a computer for controlling transfer of files between primary storage and secondary storage.

Thus, for the foregoing reasons, it is respectfully urged that the claimed invention is not obvious from the combination of Johnson and Saks for the reasons set forth herein.

U.S. Patent 5,778,420 to Ish et al

U.S. Patent 5,778,430 to Ish et al (hereinafter Ish) discloses a method and apparatus for implementation and management of a disk cache system. In response to commands received from an external source, i.e., WRITE, READ, the storage system transfers data, preferably organized as blocks, to/from direct access devices to/from the external source. The Examiner has cited column 6, line 7-13 for the proposition that Ish teaches a method and apparatus for

computer disk cache management in which hashing directories is done according to frequency of access. It is respectfully urged that this is a hindsight misinterpretation of the teachings of the references in an attempt to fill in a missing gap in the previously advanced rejections.

A review of the cited portion of the patent reveals that Ish teaches that each header in a linked chain contains a forward pointer which points to the next header in the chain and a backward pointer which points to a previous header in the list (or the hash table if the header is the first in the chain). A frequency member identifies the number of times that the particular cache line associated with the header has been accessed. However, there is no discussion or suggestion herein of selecting directories to cache using at least one of the number of files in a directory and the frequency of use of the files as set forth in the claims. Yet still further, there is no teaching or suggestion of hashing selected directories into hash table format according to at least one of size of the directory, frequency of access, and a user selected criteria.

Thus, for the foregoing reasons, it is respectfully urged that the invention as now recited in the independent claims set forth herein is clearly not anticipated by or obvious from the cited references under 35 U.S.C §102 and/or 35 U.S.C §103, and that the application should be passed to issuance.

Finally, enclosed is an Information Disclosure Statement identifying newly uncovered references and submitting them herewith which Applicants request be considered by the Examiner. No fee is necessary since the filing of this Information Disclosure Statement is timely. More specifically, Applicants became aware of the information being submitted herein from an Office Action dated March 24, 2004 issued in co-pending U.S. Application No. 09/672,421, having EMC Corporation as common Assignee.

Notwithstanding the foregoing, should the Examiner still have any comments, questions or suggestions of a nature necessary to expedite prosecution of the application or to place the case in condition for allowance, he is courteously requested to telephone the undersigned at the number listed below.

Dated: April 6, 2004

Respectfully submitted,

A. José Cortina, Reg. No. 29,733
Daniels Daniels & Verdonik, P.A.
P.O. Drawer 12218
Research Triangle Park, NC 27709
Voice 919.544.5444
Fax 919.544.5920
Email jcortina@d2vlaw.com

Enclosures

F:\CL\1301-003\Prosecution\RCE\RCE Amendment.doc